

УДК 519.8

**ОПТИМИЗАЦИЯ АЛГОРИТМА РЕШЕНИЯ
ЗАДАЧИ
О НАЗНАЧЕНИЯХ ПРИ ПОМОЩИ ТЕХНОЛОГИИ
CUDA**

Е. А. Гришкун,

*Житомирский государственный технологический университет
evgenii2081991@gmail.com*

Е. В. Квитка,

*Житомирский государственный технологический университет
kvito4ka1@gmail.com*

Данная работа посвящена разработке оптимизации уже существующего, известного алгоритма решения задачи о назначениях. В статье рассматривается применение параллельных вычислений к решению рассматриваемой задачи.

Gryshkun I.A., Kvitka K.V. Optimization of Solving the assignment problem algorithm with the help of CUDA technology

This work is dedicated to optimization development of already existing and well-known algorithm of solving assignment problems. The work examines the usage of parallel computing while solving current problem.

Ключевые слова: ЗАДАЧА О НАЗНАЧЕНИЯХ, ТЕХНОЛОГИЯ CUDA, ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ

Keywords: ASSIGNMENT PROBLEM, CUDA TECHNOLOGY, PARALLEL COMPUTING

Задача о назначениях (ЗН) [1] относится к числу немногих дискретных оптимизационных задач, для которых известны точные алгоритмы решения с полиномиальной временной сложностью.

Наиболее популярными методами решения ЗН являются венгерский метод, метод Кана-Мункреса и метод потенциалов, построены с использованием разных подходов, применяемых в

комбинаторной оптимизации, и характеризуются разной временной сложностью, не меньшей, чем $O(n^3)$, где n – порядок матрицы стоимостей.

Тем не менее при больших размерностях задачи время вычисления алгоритма становится неприемлемым, особенно при итеративном его использовании. [2]

В работе предложено метод оптимизации работы алгоритма при помощи параллельный вычислений.

Решение многих востребованных задач, таких как компьютерное моделирование, обработка видео, визуализация, распознавание образов, вычислительная биология и химия, сейсмический анализ, прогнозирование временных рядов, финансовый анализ и т.д., требует значительных затрат процессорного времени на выполнение вычислений. Эту проблему можно решить использованием технологии параллельного программирования.

Параллельное программирование применяется тогда, когда для последовательной программы требуется уменьшить время ее выполнения, или когда последовательна программа, в виду большого объема данных, перестает помещаться в память одного компьютера. Направление развития в области высокопроизводительных вычислений как раз направлено на решение этих двух задач: создание мощных вычислительных комплексов с большим объемом оперативной памяти с одной стороны и разработка соответствующего ПО с другой.

Параллельные вычисления – способ организации компьютерных вычислений, при котором программы разрабатываются как набор взаимодействующих вычислительных процессов, работающих параллельно (одновременно). Термин охватывает совокупность вопросов параллелизма в программировании, а также создание эффективно действующих аппаратных реализаций. Теория параллельных вычислений составляет раздел прикладной теории алгоритмов.

Параллельные программы могут физически исполняться либо последовательно на единственном процессоре - перемежая по очереди шаги выполнения каждого вычислительного процесса, либо параллельно - выделяя каждому вычислительному процессу

один или несколько процессоров (находящихся рядом или распределённых в компьютерную сеть).

Основная сложность при проектировании параллельных программ - обеспечить правильную последовательность взаимодействий между различными вычислительными процессами, а также координацию ресурсов, разделяемых между процессами.

Существуют различные способы реализации параллельных вычислений.

Два популярных процессора для параллельных вычислений – CPU (англ. – central processing unit) и GPU (англ. – graphics processing unit) .

Как CPU, так и GPU являются процессорами, и между ними есть много общего, однако сконструированы они были для выполнения различных задач.

Для параллельных вычислений на CPU используются технологию Parallel, которая аналогична OpenMP. Parallel – внутренняя технология, используемая в языке C#, предоставляющая поддержку параллельных циклов и областей.

Для распараллеливание на GPU используется технология CUDA.

CUDA – это архитектура параллельных вычислений от NVIDIA, позволяющая существенно увеличить вычислительную производительность благодаря использованию GPU.

CUDA - это программно-аппаратная вычислительная архитектура Nvidia, основанная на расширении языка Си, которая даёт возможность организации доступа к набору инструкций графического ускорителя и управления его памятью при организации параллельных вычислений. CUDA помогает реализовывать алгоритмы, выполнимые на графических процессорах видеоускорителей Geforce восьмого поколения и старше (серии Geforce 8, Geforce 9, Geforce 200), а также Quadro и Tesla. [3]

Результаты вычислительного эксперимента приведены в на рис. 1.

Нами предлагается использовать технологию параллельных вычислений для решения ЗН, что позволяет ускорить вычислительный процесс.

Экспериментально исследованы временные характеристики венгерского алгоритма для решения ЗН и алгоритма с применением технологии CUDA. Результаты эксперимента подтверждают, что применение параллельных вычислений для решения ЗН эффективно снижает временные затраты на решение.

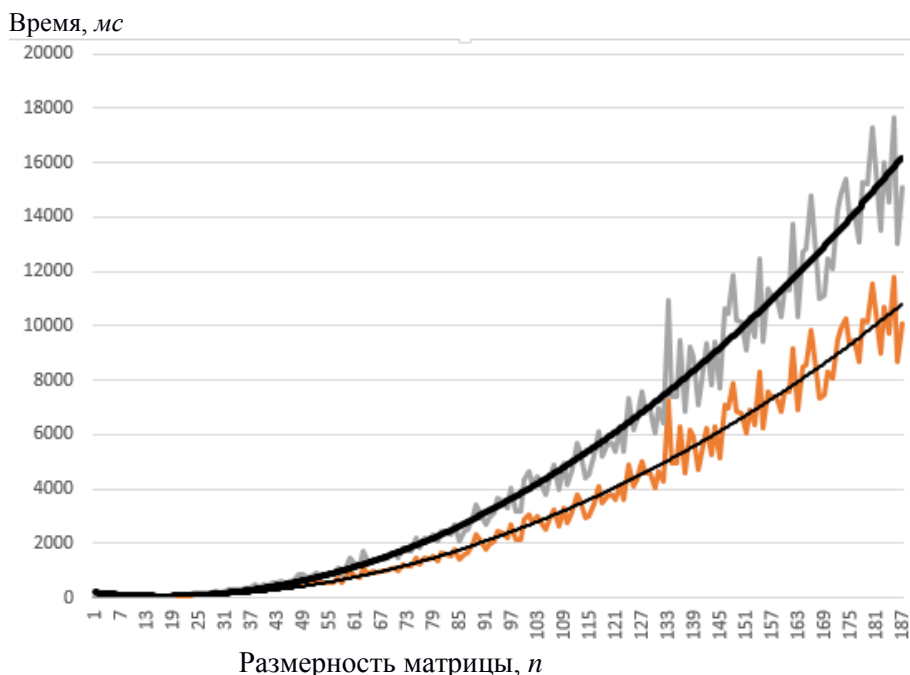


Рисунок 1 – Зависимость времени решения ЗН венгерским алгоритмом и алгоритмом с применением технологии CUDA

— - венгерский алгоритм
— - алгоритмом с применением технологии CUDA

Литература

1. Панишев А. В. Модели и методы оптимизации замкнутых маршрутов на транспортных сетях / А. В. Панишев, А. В. Морозов. – Житомир: ЖДТУ, 2014. – 324 с.

2. Маций О.Б. Рекуррентный метод решения задачи о назначениях / О.Б. Маций, А.В. Морозов, А.В. Панишев// Искусственный интеллект. – 2014. – № 2. – С. 107-118.
3. Джейсон Сандерс Технология CUDA в примерах / Джейсон Сандерс, Эдвард Кэндрот – М.: ДМК-Пресс, 2013. – 288 с.